

**EXAMINER'S AMENDMENT**

1. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. Authorization for this examiner's amendment was given in a telephone interview with David A. Dagg, Reg.# 37,809 on 07/16/2009.

3. **This listing of claims will replace all prior versions and listings of claims in the application:**

1. (amended) A method embodied in a computer system for monitoring memory usage of an agent executing in said computer system, wherein said agent is a software agent comprising at least a portion of a software application, said method comprising:

starting a resource tracking application for monitoring memory usage of said agent;

creating a computer-readable data structure for storing information about said agent;

identifying a process that is currently running on said computer system, and with which said agent is operatively associated, including identifying a plurality of threads of said process that said agent is running on;

determining, by said resource tracking application, memory usage data for said agent, wherein said memory usage for said agent is determined based on a total memory usage of said plurality of threads;

storing said memory usage data in said data structure;

determining, responsive to said memory usage data stored in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold; and

displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier uniquely associated with said agent, a recommended solution to address said exceeding of said predetermined maximum memory usage threshold by said agent, and an execute solution user interface object, wherein selecting of said execute solution user interface object by a user causes said recommended solution to be automatically performed.

2. (original) The method of claim 1, wherein said computer-readable data structure is a hash table.

3. (previously presented) The method of claim 2, wherein said determining said process that is currently running further comprises determining that said process is a non-hypertext transport protocol (non-HTTP) process comprising at least one of the set consisting of an electronic mail task, a Web conferencing task, and a customer developed task.

4. (currently amended) The method of claim 3 wherein said determining said process that is currently running further includes:

creating a thread list for monitoring said plurality of threads ~~associated with said agent;~~

~~identifying at least one thread associated with said at least one identified thread to produce~~ producing at least one determined allocation for said at least one of said plurality of identified threads; and

computing a peak memory usage for said agent using said at least one determined allocation;

thereby monitoring memory usage by said agent.

5. (original) The method of claim 4 further comprising:

comparing said peak usage for said agent to a plurality of peak usages associated with a like plurality of other agents executing in said system.

6. (original) The method of claim 5 further comprising:

computing statistics on said agent and said plurality of other agents; and

ranking said agent against said plurality of other agents based on said peak usage to produce a ranked list.

7. (previously presented) The method of claim 6 further comprising:

displaying said ranked list to said user.

8. (previously presented) The method of claim 2, wherein said determining said process that is currently running further comprises determining that said process is an HTTP process comprising an HTTP task.

9. (currently amended) The method of claim 8, wherein said determining said process that is currently running further includes:

wherein said identifying said plurality of threads of said process said agent is running on includes identifying HTTP threads operating in said system to produce identified threads, each of said ~~identified~~ HTTP threads further having one of a plurality of agent types associated therewith, at least one of said plurality of agent types including said agent, said agent capable of having agent threads associated therewith;

generating an agent thread list for facilitating identification of said plurality of agent types by storing information associated therewith; and

identifying which of said plurality of agent types is operating on each of said HTTP threads;

~~associating those of said agent threads said agent is running on together to produce a related agent set;~~

~~determining memory usage for each thread in said related agent set; and~~

~~combining said memory usage for each thread in said related agent set to produce a total memory consumption for said agent;~~

~~thereby monitoring memory usage by said agent.~~

Art Unit: 2195

10. (currently amended) The method of claim 9 further comprising:

comparing said total memory consumption for said agent to a like plurality of total memory consumptions associated with others of said plurality of agent[s] types.

11. (original) The method of claim 10 further comprising:

computing statistics on said total memory consumption for said agent and each one of said plurality of total memory consumptions; and

ranking said agent against said others of said plurality of agent types using said total memory consumption and said plurality of total memory consumptions, respectively, to produce a ranked list.

12. (previously presented) The method of claim 11 further comprising:

displaying said ranked list to said user.

13. (original) The method of claim 1, further comprising:

determining if said agent is running before determining said memory usage.

14. (original) The method of claim 13, further comprising:

determining if said agent is expired; and

processing said information if said agent is expired.

15. (original) The method of claim 1, wherein said data structure further includes information about a plurality of other software agents.

16. (previously presented) The method of claim 15, further comprising:

establishing said maximum memory usage threshold.

17. (original) The method of claim 16, further comprising:

terminating said software agent and those of said plurality of other software agents exceeding said threshold.

18. (currently amended) A computer program product including a memory having machine-readable instructions stored thereon for, when executed, causing a processor to perform a method for identifying memory usage information associated with a software agent operating in a computer system, said instructions comprising:

instructions for initiating a resource tracking application for monitoring memory usage of said agent;

instructions for generating a computer-readable data structure residing in computer-accessible memory for storing memory usage data associated with said agent;

instructions for determining, by said resource tracking application, said memory usage data, wherein said memory usage data is determined based on a total memory usage of a plurality of threads of a process said agent is running on;

instructions for storing said memory usage data in said computer-readable data structure;

instructions for determining, responsive to said memory usage data stored in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold; and

instructions for displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier uniquely associated with said agent, a recommended solution to address said exceeding of said predetermined maximum memory usage threshold by said agent, and an execute solution user interface object, wherein selecting of said execute solution user interface object by a user causes said recommended solution to be automatically performed.

19. (previously presented) The computer program product of claim 18, said instructions further comprising:

instructions for storing memory usage information about a plurality of other software agents;

instructions for processing said memory usage information associated with said agent and said memory usage information about said plurality of other software agents; and

instructions for generating a rank order list including said information about said agent and said information about said plurality of other software agents.

Art Unit: 2195

20. (currently amended) A computer system for monitoring operation of an agent executing in said computer system, wherein said agent is a software agent comprising at least a portion of a software application, said computer system comprising:

a processor for executing instructions stored in a computer readable memory, said instructions when executed performing the steps of:

starting a resource tracking application for monitoring memory usage of said agent;

creating a computer-readable data structure for storing information about said agent;

identifying a process that is currently running on said computer system, and with which said agent is operatively associated, including identifying a plurality of threads of said process said agent is running on;

determining, by said resource tracking application, memory usage data for said agent, wherein said memory usage for said agent is determined based on a total memory usage of said plurality of threads;

storing said memory usage data in said data structure;

determining, responsive to said memory usage data stored in said data structure, that said memory usage of said agent exceeds a predetermined maximum memory usage threshold; and

displaying, responsive to said determination that said memory usage of said agent exceeds said predetermined maximum memory usage threshold, a system administrator user interface, said system administrator user interface including an agent identifier uniquely associated with said agent, a recommended



Art Unit: 2195

solution to address said exceeding of said predetermined maximum memory usage threshold by said agent, and an execute solution user interface object, wherein selecting of said execute solution user interface object by a user causes said recommended solution to be automatically performed.

4. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

### ***Conclusion***

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ABDULLAH AL KAWSAR whose telephone number is (571)270-3169. The examiner can normally be reached on 7:30am to 5:00pm, EST.

6. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng Ai T. An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Art Unit: 2195

7. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./  
Supervisory Patent Examiner, Art Unit 2193

/Abdullah-Al Kawsar/  
Examiner, Art Unit 2195